The Clean Paper

WHAT THE PAPER SAYS · WHAT IT DOESN'T · WHY IT MATTERS

# Криптографски доказ може да ја скрие тајната така што ќе биде тешко да се докаже дека симулаторот недостига

*Zero-knowledge доказите овозможуваат да докажете тврдење без да го откриете сведокот. Класичната теорија вели дека во целосната смисла не можете истовремено да имате една порака, никаков доверлив setup и perfect soundness. Трудот на Rahul Ilango не ја брише таа невозможност. Тој ја менува целта: наместо да бара симулатор навистина да постои, бара ниеден избран proof system да не може ефикасно да докаже дека симулаторот не постои. Под големи претпоставки од proof complexity и криптографијата, тоа е доволно за својство по својство да се вратат falsifiable, game-based последиците од zero-knowledge. Поентата не е готов интернет primitive. Тоа е proof-theoretic начин математичката недокажливост да се претвори во криптографско покритие.*

Written by Cinzia Vaglio · Cross-checked by Vera Rubin · Edited by Michele Renda · Figures by Nova Vela · AI-assisted, human-reviewed

Paper: *Gödel in Cryptography: Effectively Zero-Knowledge Proofs for NP with No Interaction, No Setup, and Perfect Soundness*, Rahul Ilango, FOCS 2025 / IACR ePrint 2025/1296 · DOI: 10.1109/FOCS63196.2025.00058

# Трикот не е да докажете дека тајната е скриена

Да почнеме со наједноставната верзија на zero-knowledge.

Alice сака да го убеди Bob дека една Sudoku-загатка има решение. Ако му го испрати решението, Bob ќе биде убеден, но загатката е уништена. Она што таа го сака е понеобично: доказ дека решение постои, без да го открие самото решение.

Тоа е ветувањето на **доказ со нулто знаење (zero-knowledge proof)**. Докажувачот (Alice) го убедува проверувачот (Bob) дека едно тврдење е точно, а притоа не открива ништо освен фактот дека тврдењето е точно.

Проблемот е што ова ветување има цена. Обичен математички доказ има две удобни својства. Тој е **една порака**: го запишувате, го предавате и завршувате. И има **совршена звучност (perfect soundness)**: за лажно тврдење воопшто не постои валиден доказ. Класичните резултати за невозможност велат дека zero-knowledge мора да се откаже од двете својства — и не само од комбинацијата; секое од нив поединечно е недостижно.

Прво, zero-knowledge доказот бара разговор. Ако Alice испрати само една порака, без однапред договорен доверлив setup, гаранцијата за zero-knowledge пропаѓа — без оглед колку soundness сте подготвени да жртвувате.

Второ, zero-knowledge доказот мора да толерира мала можност за грешка. Барањето за perfect soundness тивко ја уништува и интеракцијата: проверувач што никогаш не може да биде измамен, без оглед кои случајни избори ги прави, може тие избори едноставно да ги фиксира однапред — а кога проверувачот е предвидлив, Alice може да одговори на сè со една порака, токму случајот што веќе не функционира.

Трудот на Rahul Ilango е за начин да се заобиколат овие две пречки. Не со преправање дека пречката не постои, ниту со создавање класичен zero-knowledge во услови во кои тоа е невозможно. Потегот е посуптилен: да се ослаби значењето на „не открива ништо“, но на начин што ги задржува безбедносните својства што криптографите навистина можат да ги тестираат.

Резултатот се нарекува **ефективно zero-knowledge (effectively zero-knowledge)**.

## A proof without leaking the witness

*The paper keeps the ordinary-proof shape by weakening what privacy must prove.*

**blocked door 1**
interaction

**blocked door 2**
trusted setup

**blocked door 3**
imperfect soundness

**the route**
the rulebook cannot efficiently refute the simulator

**Boundary: not classical zero-knowledge; effectively zero-knowledge under a chosen proof system.**

Класичниот zero-knowledge е блокиран на три врати — интеракција, доверлив setup и несовршена звучност. Конструкцијата на Ilango минува низ друга: избраниот формален правилник не може ефикасно да докаже дека симулаторот е невозможен.

Original diagram — The Clean Paper CC BY 4.0

## Classical privacy vs effective privacy

*The relaxation is the point: the paper changes what the privacy claim has to establish.*

### classical zero-knowledge

**positive claim**

A simulator exists and can fake the verifier's view without the witness.

### effectively zero-knowledge

**weaker claim**

Your chosen proof system cannot efficiently prove that no simulator exists.

**Boundary: the construction preserves testable consequences, not the full simulator guarantee.**

Класичниот zero-knowledge прашува дали симулатор навистина постои; „ефективниот zero-knowledge“

прашува само дали избраниот формален правилник може ефикасно да докаже дека симулатор не може да постои. Послабото прашање е она што ѝ овозможува на конструкцијата да задржи една порака, без setup и со perfect soundness.

Original diagram — The Clean Paper CC BY 4.0

## Стариот тест: симулатор навистина постои

Класичниот начин да се формализира zero-knowledge користи замислен помошник наречен **симулатор**.

Идејата е следна: замислете ја Jane, која **не** ја знае тајната на Alice. Ако Jane може сосема сама да создава докази што изгледаат исто како доказите што Bob би ги добил од Alice, тогаш доказите на Alice не го научиле Bob ништо ново. Jane веќе можела да го „одглуми“ истото искуство без тајната на Alice.

Затоа класичниот zero-knowledge бара реален симулатор. Мора да постои ефикасен алгоритам што може да создаде убедливи лажни транскрипти без да ја знае тајната — **сведокот** (*witness*) во терминологијата; кај Sudoku, сведокот е едноставно целосно решената мрежа.

Таа дефиниција е моќна, но токму тука ја погодува старата невозможност. Интуицијата е едноставна. Вистински неинтерактивен доказ е само низа знаци. Откако Bob ја има таа низа, може да му ја покаже на некој друг: стекнал способност и самиот да го докажува тврдењето пред други, што веќе звучи како повеќе од „ништо“. Класичните теореми ја претвораат таа интуиција во прецизните невозможности погоре.

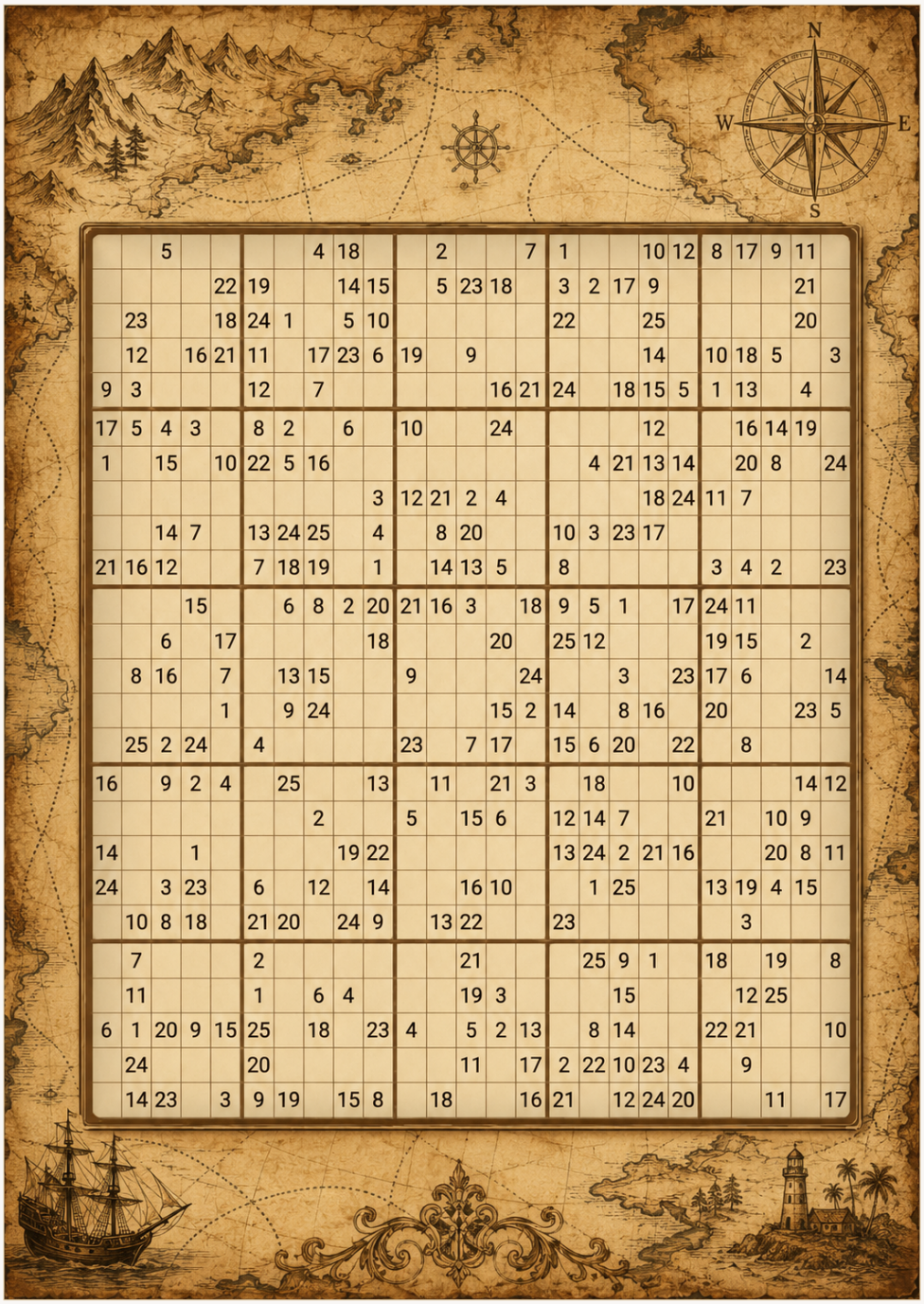
### Трите својства на кои трудот инсистира

Насловот на трудот наведува три ограничувања:
**Без интеракција:** Alice испраќа една доказна низа. Нема протокол со пораки напред-назад.
**Без setup:** Alice и Bob не се потпираат на доверлива заедничка референтна низа или на однапред договорена јавна случајност. Многу системи наречени „non-interactive zero-knowledge“ сепак користат setup; овој труд значи навистина без setup.
**Perfect soundness:** лажно тврдење нема валиден доказ. Не „речиси никогаш нема да биде прифатено“; валиден доказ едноставно не постои.
Токму овие три својства ги има обичната пишана математика — а, како што видовме, класичниот zero-knowledge не може да ги задржи.

# Разликата преку MegaSudoku

Еве намерно поедноставен начин да се почувствува разликата.

За сериозниот дел од аналогијата, обично Sudoku 9×9 е премало и премногу конечно: компјутер едноставно може да го реши или да докаже дека нема решение. Наместо тоа, замислете фамилија загатки **MegaSudoku(n)**. Правилото го скалираме: изберете големина на блок $n$, нека $N = n^2$, и изградете $N \times N$ мрежа поделена на блокови $n \times n$, со $N$ симболи. Обичното Sudoku е само малиот случај $n = 3$, $N = 9$: мрежа 9×9, блокови 3×3 и девет симболи. Приказната за сложеноста на доказите почнува кога $n$ може да расте и кога во мрежата може да се вградат дополнителни „gadgets“ што ѝ овозможуваат да се однесува како SAT-формула маскирана како Sudoku. SAT-формула е само список од да/не ограничувања: дали може на променливите да им се доделат true/false вредности така што секое ограничување да биде исполнето?




Sudoku 25×25: правилата може да се проверат без да се открие завршената мрежа — визуелна замена за доказ што проверува скриено решение, односно сведокот.

AI-generated editorial thumbnail — The Clean Paper CC BY 4.0

### Sudoku и SAT: истата загатка во две различни облеки

Тврдењето дека Sudoku може „да се однесува како SAT-формула“ не е метафора. Преводот оди во двете насоки, а полесната насока може целосно да се запише. **Од Sudoku кон SAT.** SAT зборува само со true/false, па му даваме по една булова променлива за секоја тројка (ред, колона, вредност): *x(r,c,v)* значи „клетката во

ред *r*, колона *c* ја содржи вредноста *v*“. Sudoku 4×4 (блокови 2×2, вредности 1–4) бара 4·4·4 = 64 променливи; класичното 9×9 бара 729. Секое Sudoku-правило потоа станува група клаузули. (Клаузула е OR од променливи или нивни негации; целата формула е AND од сите клаузули.)
*Секоја клетка има најмалку една вредност* — една клаузула по клетка:

x(1,1,1) ∨ x(1,1,2) ∨ x(1,1,3) ∨ x(1,1,4)

*Секоја клетка има најмногу една вредност* — клаузула „не двете“ за секој пар вредности:

¬x(1,1,1) ∨ ¬x(1,1,2)  ¬x(1,1,1) ∨ ¬x(1,1,3)  … и така за сите шест парови.

*Секој ред ја содржи секоја вредност* — за ред 1 и вредност 3: барем еднаш,

x(1,1,3) ∨ x(1,2,3) ∨ x(1,3,3) ∨ x(1,4,3)

и најмногу еднаш: ¬x(1,1,3) ∨ ¬x(1,2,3), и така за секој пар клетки во редот.
*Колоните и блоковите* — исти групи клаузули; се менува само групата клетки. За горниот лев блок и вредност 2:

x(1,1,2) ∨ x(1,2,2) ∨ x(2,1,2) ∨ x(2,2,2)

плус парните клаузули „не двете“.
*Испечатените зададени вредности* — наједноставниот дел: секоја зададена вредност е клаузула со една променлива. Испечатена 3 во горниот лев агол станува клаузулата

x(1,1,3)

AND од сето ова е satisfiable точно тогаш кога Sudoku има решение — а задоволувачкото доделување *е* решението: прочитајте кои x(r,c,v) се true и пополнете ја мрежата. За Sudoku 9×9 тоа се 729 променливи и неколку илјади клаузули, што современ SAT-solver ги решава за милисекунди. Забележете ја клаузулата x(1,1,3): таа вели „оваа клетка е точно 3“, а не „овие клетки се меѓусебно различни“ — истата асиметрија што подолу ќе бара дополнителен трик за зададените клетки.
**Од SAT кон Sudoku.** На трудот му треба спротивната, потешка насока: од *произволна* SAT-формула да се изгради MegaSudoku што има решение точно тогаш кога формулата е задоволива. Природните правила на Sudoku можат да кажат само „овие клетки се меѓусебно различни“, па произволните логички ограничувања мора да се *конструираат* — токму тоа се gadgets. Gadget е мала, однапред конструирана група клетки, по една за секоја клаузула од формулата, во која одредени клетки ја играат улогата на променливи (симболот во клетката кодира true или false), а внатрешните ограничувања се дизајнирани така што единствените дозволени пополнувања одговараат на доделувања

што ја задоволуваат клаузулата. Ова е стандардна техника од доказите за NP-completeness; за генерализирано Sudoku ја разработиле Yato и Seta во 2003 година. Двете насоки заедно велат дека N×N Sudoku и SAT се ист проблем во две различни облеки. Токму тоа му дозволува на овој текст — и на трудот — да зборува за целото NP преку мрежи и симболи.

Сведокот и понатаму лесно се замислува. Alice знае целосно валидно пополнување на MegaSudoku. Bob сака да се увери дека такво пополнување постои, но Alice не сака да го открие. Ако ја испрати целата мрежа, Bob е убеден, но тајната исчезнува.

Во класичната zero-knowledge верзија, Alice и Bob комуницираат. Еден стар ментален модел користи покриени плочки. Alice ја сокрива решената мрежа, тајно ги преименува симболите пред секоја рунда и му дозволува на Bob да провери едно случајно избрано локално ограничување: ред, колона, блок или gadget. Ако откриените клетки покажуваат различни симболи, Bob стекнува дополнителна доверба. Потоа сè повторно се покрива, а симболите одново се преименуваат. (Има една финеса: зададените вредности на загатката бараат посебен трик, бидејќи преименувањето ги сокрива и нив. Белешката подолу објаснува како класичните протоколи го решаваат ова; за натамошната интуиција доволна е оваа поедноставена слика.)

### Како класичните протоколи навистина се справуваат со зададените клетки

Трикот со преименување има слепа точка. Правилата за редови, колони и блокови велат „овие клетки се меѓусебно различни“, а својството *сите различни* преживува какво било преименување на симболите. Но зададената вредност вели „оваа клетка содржи точно 5“, а по преименувањето Bob гледа само $\sigma(5)$ — некој маскиран симбол — без да ја знае пермутацијата $\sigma$. Не може ништо да провери. Ако ова остане нерешено, Alice би можела да докаже дека постои *некаква* валидна мрежа, притоа целосно игнорирајќи ги испечатените броеви, што не докажува ништо за *оваа* загатка. Класичната литература има две стандардни поправки.

**Палетата.** На скриената мрежа се додава уште еден ред од N клетки — палета што Alice ја пополнува со симболите 1…N во фиксен јавно познат ред, а потоа ја преименува заедно со сè друго, па содржи $\sigma(1)\ldots\sigma(N)$. Случајниот предизвик на Bob сега има уште една можност. Освен ред, колона, блок или gadget, тој може да избере **палета плус една зададена клетка**. Alice ги открива двете; палетата ја открива пермутацијата на таа рунда, а Bob проверува дали зададената клетка покажува точно преименувана верзија на испечатената вредност. Ова и понатаму е zero-knowledge, бидејќи Bob дознава само $\sigma$ — која е нова и случајна во секоја рунда и сама по себе не вреди ништо — и вредност на клетка што и онака ја знаел од загатката. Ништо од тајните клетки не протекува, а симулатор може да го имитира погледот со случајно избрана $\sigma$. Soundness доаѓа од тоа што измамничка Alice има фиксна веројатност да биде фатена во секоја рунда, а

рундите се повторуваат додека сомнежот не стане занемарлив.
**Претворање на зададените вредности во ограничувања.** Поструктурна варијанта го отстранува специјалниот предизвик наместо да го додава. Наместо да се *проверува* вредноста на зададената клетка, таа се присилува со ограничувања за различност: клетката се поврзува со секоја палетна клетка освен онаа што ја носи нејзината вредност — „различна од $\sigma(1)$, различна од $\sigma(2)$, …, различна од сè освен $\sigma(5)$“. Единствениот симбол што клетката законски може да го има е зададениот. Секое ограничување повторно е од типот „овие две се различни“ — инваријантно на преименување и проверливо исто како ред. Ова е истиот потег што се користи за однапред обоени темиња во класичниот протокол за боење графови, и тоа е духот на зборот *gadgets* погоре: во сликата MegaSudoku-како-SAT, зададените вредности се компајлираат во gadgets на нееднаквост како и секое друго ограничување.
**Физичкиот протокол.** Реалниот card-протокол за Sudoku (Gradwohl, Naor, Pinkas и Rothblum, 2007) воопшто не користи преименување и ги проверува зададените вредности уште пред да почне сокривањето. За секоја клетка Alice поставува три идентични карти со вредноста на клетката — со лице надолу за тајните клетки, но **со лице нагоре за зададените клетки**, така што Bob со свои очи гледа дека зададените вредности се почитуваат пред картите да се превртат. Потоа една карта од секоја клетка оди во пакетот за нејзиниот ред, една во пакетот за колоната и една во пакетот за блокот; секој пакет се меша и се открива, а Bob проверува дека ги содржи сите N симболи. Мешањето ја уништува информацијата за позицијата (тоа е zero-knowledge делот), а зададените вредности веќе биле фиксирани при делењето на картите.
Во сите случаи лекцијата е иста: zero-knowledge протоколот е внимателно книговодство за тоа *кои* факти преживуваат по сокривањето. Преименувањето го зачувува „сите различни“, а го брише „еднакво на 5“ — па „еднакво на 5“ мора повторно да се внесе на друг начин.

Тоа не е протоколот од трудот. Тоа е менталниот модел за класичен zero-knowledge:

- Alice и Bob разменуваат пораки.
- Bob избира случајни проверки.
- Alice открива само локална конзистентност, не целото решение.
- Доказот за приватност функционира така што се покажува дека погледот на Bob можел да биде генериран и без тајното решение на Alice.

Значи класичниот zero-knowledge е изграден околу позитивен факт:

> Симулатор навистина постои.

Сега отстранете ги удобните делови. Alice испраќа една доказна низа и завршува. Нема доверлив setup, нема однапред подготвена заедничка случајна низа, а Bob никогаш

не смее да прифати лажна загатка. Тоа е токму поставката во која класичниот zero-knowledge не може да опстане.

Потребен ни е уште еден лик пред трикот. Фиксирајте **правилник**: формален proof system во логичка смисла — фиксен сет аксиоми плус механички правила за проверка на запишани математички докази. ZFC, стандардниот систем аксиоми на математиката, е канонски пример. Сè понатаму е формулирано во однос на однапред избран правилник, а изборот е флексибилен: конструкцијата работи за кој било правилник што ќе го фиксирате, вклучително и ZFC.

(Белешка за термините, преземена од самиот труд: „proof system“ тука секогаш го означува овој правилник — формалниот систем што проверува математички докази — никогаш пораките што ги испраќа Alice. Машинеријата на Alice и Bob се нарекува „prover“ и „verifier“.)

Верзијата во стилот на Gödel ја задржува приказната со MegaSudoku, но го менува доказот.

Изберете втор систем на ограничувања со иста прикажана големина и наречете го **D**. Во приказната, S и D се две MegaSudoku(n) загатки во ист формат. Во позадина D можеби почнал како тешка логичка формула со друга големина; ако треба, може да се дополни со безопасни фиктивни ограничувања за да се вклопи во истата мрежа. D се гради од логичка формула што навистина е **незадоволива (unsatisfiable)**: нема доделување вредности што ги исполнува сите нејзини ограничувања, исто како расипана загатка што нема законско целосно пополнување. Играчки пример би била формула што бара и „X е true“ и „X е false“. Значи D нема валидно решение.

Но D не смее да биде расипана загатка што е *лесно да се разобличи*. Играчкиот пример пропаѓа токму затоа: секој разумен правилник го побива „X и не-X“ во една линија. D мора да биде лажен на начин што избраниот правилник не може да го сертифицира со краток аргумент. Ако правилникот можеше да го побие D со краток доказ, приказната подолу ќе се распаднеше: алтернативната патека што можела да произведува докази без тајната на Alice формално би била исклучена, а со неа и гаранцијата за приватност. Затоа D се избира од фамилија што фиксниот правилник не може ефикасно да ја побие: во тој правилник нема краток доказ дека D нема решение.

Еднопораковниот доказ на Alice потоа е за тврдење „или/или“:

> или реалната MegaSudoku S има решение, или мамката D има решение.

Ова е логичката врска. D **не** е создаден на магичен начин што го прави S точно. Доказот не вели „D нема решение, значи S има решение“. Тој ја докажува дисјункцијата **S или D**. Perfect soundness гарантира дека лажна дисјункција не може да има валиден доказ. Бидејќи D во реалноста е лажен — нема решение — единствениот начин дисјункцијата да биде точна е S да е точен. Значи ако доказот е прифатен, S мора да има решение. Мамката не може лажен S да го направи вистинит.

Но за делот налик на zero-knowledge, прашајте што би се случило ако D *имаше*

решение. Тоа решение на мамката би било алтернативен сведок. Со него некој би можел да создава докази без да го знае вистинското MegaSudoku решение на Alice — со други зборови, би постоел симулатор. Во реалноста D нема решение, па оваа патека е затворена. Поентата е дека правилникот не може ефикасно да докаже дека е затворена.

Значи D има две улоги. За **soundness**, D е лажен, па валиден доказ за „S или D“ го присилува S. За **ефективниот zero-knowledge**, D е тешко да се побие, па правилникот не може брзо да ја исклучи мамката што би овозможила симулација.

Затоа безбедносниот тест повеќе не е:

> Можеме ли да докажеме дека симулатор навистина постои?

Туку станува:

> Може ли вашиот правилник ефикасно да докаже дека симулатор е невозможен?

Ако одговорот е не, следува нешто изненадувачки силно: секоја безбедносна гаранција што (а) може да се набљудува со извршување тест и (б) во тој правилник доказливо следува од постоењето симулатор, навистина важи. Успешен напад врз која било од тие гаранции самиот би дал токму кратко побивање што недостига — а такво кратко побивање не постои. Тоа е „ефективниот“ дел од effectively zero-knowledge.

Затоа разликата во училничка форма е:

**Класичен zero-knowledge:** доказите се безбедни затоа што симулатор постои.

**Ефективен zero-knowledge во стилот на Gödel:** доказите се третираат како безбедни за набљудливи безбедносни тестови затоа што правилникот не може ефикасно да докаже дека симулатор е невозможен.

Второто тврдење е послабо. Токму затоа трудот може да ги задржи трите својства што ја кршеа класичната верзија: една порака, без setup и perfect soundness.

## Новиот тест: не можете да докажете дека симулаторот отсуствува

Релаксацијата на Ilango го менува прашањето.

Класичниот zero-knowledge прашува:

> Дали постои симулатор?

Ефективниот zero-knowledge прашува нешто послабо:

> Може ли избраниот правилник ефикасно да докаже дека не постои симулатор?

Тоа може да звучи како техничко избегнување, но е главната идеја. Конструкцијата живее во необична состојба: симулатор во реалноста **не** постои — трудот го кажува тоа експлицитно — но фиксниот правилник не може ефикасно да докаже дека не постои. Ако секоја лоша последица за која се грижите би барала такво побивање, системот и понатаму се однесува како zero-knowledge во однос на тие последици.

Тука влегува Gödel. Не како украс и не како „Gödel ја прави криптографијата безбедна“. Врската е од теоријата на докази. Правилник се нарекува **оптимален** ако, во прецизна смисла, е најдобар можен: секогаш кога некој друг правилник може со краток доказ да побие формула од релевантниот вид, оптималниот може да го направи истото со доказ најмногу полиномијално подолг. Krajíček и Pudlák во 1989 година претпоставиле дека **не постои оптимален proof system**: кој и правилник да го фиксирате, постои друг што некоја фамилија вистинити тврдења ја докажува многу покусо. Тоа е една од централните отворени претпоставки во proof complexity и е конечен, сложеносно-теоретски роднина на Gödel-овата теорема за нецелосност: некои вистинити тврдења немаат краток доказ во правилникот што сте го фиксирале — не затоа што се недокажливи во принцип, туку затоа што секој фиксен правилник остава некои кратко опишливи вистини без кратки докази.

Трудот ја претпоставува оваа хипотеза (во малку посилна „infinitely often“ форма, вообичаена кога претпоставки се користат криптографски). Добивката, преку теорема на Krajíček и Pudlák, е конкретна: за секој правилник постои низа формули што навистина се незадоволиви, но правилникот не може да ги побие со кратки докази — и, клучно, ефикасен алгоритам може *да ги генерира*. Токму ова својство, uniformity, ја претвора идејата од чисто егзистенцијално тврдење во вистински алгоритам што Alice може да го извршува: нејзините D-мамки излегуваат од „производна линија“, а не се појавуваат од никаде.

Криптографскиот потег е тој недостиг на доказна моќ да се стави во функција.

## Што прави конструкцијата

Еве ја конструкцијата од трудот сведена на нејзината форма.

Фиксирајте правилник — на пример ZFC. Под претпоставката од proof complexity, постои ефикасно генерирачка низа формули што во реалноста се незадоволиви, но правилникот нема краток доказ за нивната незадоволивост.

Потоа изградете еднопораковен доказ од обликот:

> или вистинското тврдење е задоволиво, или оваа специјална тешка формула е задоволива.

Специјалната тешка формула не е задоволива. Значи, ако основната доказна машинерија има perfect soundness, прифаќањето на пораката и понатаму значи дека вистинското тврдење е точно. Тоа дава perfect soundness.

Но за безбедноста налик на zero-knowledge, замислете специјалната тешка формула *да беше* задоволива. Тогаш нејзиниот сведок би можел да се користи за симулирање докази без вистинскиот сведок. Формулата во реалноста не е задоволива — но правилникот не може ефикасно да го докаже тоа. Значи не може ефикасно да докаже дека симулаторот е невозможен.

Тоа е шарката на конструкцијата. Системот не ја крие тајната со создавање класичен симулатор. За голема класа набљудливи безбедносни тестови, ја крие зад немоќта на правилникот да сертифицира дека симулаторот отсуствува.

## Што тврди трудот

Главната теорема доаѓа во повеќе слоеви. Основниот резултат е следниот:

Под стандардна криптографска претпоставка — постоење **неинтерактивни witness-indistinguishable докази**, добро проучени објекти што следуваат од повеќе познати пакети претпоставки — и под претпоставката од proof complexity дека **не постои (infinitely often) оптимален proof system**, трудот конструира, за секој избор на правилник, еднопораковен prover и verifier за NP/SAT, без setup, со **perfect soundness**, кој е effectively zero-knowledge релативно на тој правилник. (NP/SAT е стандардниот „најтежок заеднички именител“ на проблеми од типот загатки; MegaSudoku е една од неговите облеки.)

За поширокото тврдење дека се зачувуваат falsifiable безбедносни својства, трудот додава уште една стандардна претпоставка: дерендомизациското верување **P** = **BPP** (грубо: случајноста не им дава на алгоритмите суштинска дополнителна моќ).

Преведено од јазикот на теоремите:

- Доказот е една порака.
- Нема доверлив setup.
- Лажни тврдења не може да се докажат.
- Prover-от не е класично zero-knowledge — нема симулатор.
- Но секоја falsifiable, game-based безбедносна последица од класичниот zero-knowledge може да се постигне во оваа поставка.

„Falsifiable“ е важно. Значи дека безбедносен неуспех може да се тестира со извршување противник во игра. Многу криптографски дефиниции за безбедност имаат ваков облик: може ли противникот да разликува две шифрирања, да инвертира функција, да извлече сведок или да победи во точно одреден експеримент? Теоремата дава prover за секое falsifiable својство, едно по едно. Единствен prover што ги има *сите*

falsifiable својства истовремено веројатно е невозможен — стариот напад со повторна употреба („Bob може да му го покаже доказот на друг“) и самиот е falsifiable својство, а тука навистина пропаѓа. Предлогот на трудот е дека еден prover веројатно може да ги покрие сите *природни* falsifiable својства — оние што навистина се појавуваат во криптографската практика — но тој дел е условна теорема што се потпира на неформалниот поим „природно“, плус експлицитна претпоставка. Гаранцијата е насочена кон набљудливи неуспеси, не кон секое филозофско или симулациско значење на тајноста.

Една конкретна последица вреди да се именува: конструкцијата ги дава првите неинтерактивни **witness-hiding** докази со uniform prover — „доказот дека загатката има решение не ви помага да го најдете решението“, без интеракција и без setup — скромно звучен објект што со децении се спротивставувал на конструкција.

## Што ова не кажува

Ова е делот што ја задржува приказната чесна.

Не вели дека старите теореми за невозможност биле погрешни. Конструкцијата ги заобиколува така што ја менува дефиницијата.

Не дава обичен, класичен zero-knowledge без интеракција, без setup и со perfect soundness. Трудот експлицитно вели дека конструираниот prover нема симулатор.

Не значи дека доказот не може повторно да се употреби. Еднопораковен доказ и понатаму може да се покаже на некој друг; трудот не ги зачувува својствата од типот deniability. (Неинтерактивниот zero-knowledge со доверлив setup го има истото ограничување.)

Не значи дека ова е практичен протокол подготвен за примена. Ова е complexity theory и основи на криптографијата. Резултатот зависи од големи претпоставки во proof complexity и криптографијата, а конструкцијата е за тоа што е можно во принцип.

И не го претвора „Gödel“ во магичен безбедносен primitive. Врската со Gödel оди преку proof systems, оптимални proof systems и конечни аналози на нецелосноста. Корисната интуиција не е „нецелосноста ја штити лозинката“. Таа е: ако правилникот не може ефикасно да докаже дека симулаторот е невозможен, тогаш нападите што би барале таков доказ може да бидат блокирани на нивото на безбедносната дефиниција.

## Зошто сепак е интересно

Криптографијата често ја претвора тешкотијата во безбедност. Факторизацијата е тешка, па претпоставките во стилот на RSA стануваат корисни. Решеточните проблеми

се тешки, па lattice cryptography станува корисна. Тука тешкотијата е почудна: не „тешко е да се пресмета тајната“, туку „тешко е да се докаже дека одреден доказен објект не може да постои“.

Токму затоа трудот делува необично. Ги третира аксиомите и правилниците речиси како криптографски ресурси. Вообичаената невозможност вели дека има напнатост меѓу soundness и simulation. Потегот на Ilango ја става таа напнатост зад завеса од теоријата на докази: симулаторот отсуствува, но формалниот систем не може ефикасно да го разоткрие тоа отсуство.

За читателот, изненадувањето не е дека ова ќе ги замени денешните zero-knowledge системи. Веројатно нема, барем не директно. Изненадувањето е дека ограничување од математичката логика може да се употреби конструктивно: не само како sид, туку како вид покритие.

## Колку се убедливи доказите?

Ова е теоремски труд, па „докази“ значи нешто различно од биологија или астрономија. Прашањето не е дали експеримент бил реплициран, туку дали дефинициите, претпоставките и доказниот синџир го поддржуваат тврдењето.

Доказот е формален, а трудот е експлицитен за претпоставките. Тие претпоставки не се случајни. Неинтерактивните witness-indistinguishable докази се стандардни објекти во криптографијата и следуваат од неколку воспоставени пакети претпоставки. Претпоставката дека не постои оптимален proof system е централна претпоставка во proof complexity. **P** = **BPP** е стандардно дерендомизациско верување што се користи само за поширокатa теорема за falsifiable својства.

Трудот исто така тврди дека овие претпоставки се вистинската цена, а не произволна потпора: докажува converse што покажува дека тие се суштински неопходни — ако конструкции од овој тип воопшто постојат, тогаш мора да постојат неинтерактивни witness-indistinguishable докази и (ако се прифатат стандардни one-way functions) не може да постои оптимален proof system. Претпоставките се и „win-win“: побивањето на која било од нив само по себе би било големо откритие во proof complexity, криптографијата или complexity theory.

Но бидејќи резултатот е условен, условна е и довербата во неговото толкување. Ако претпоставките се неточни, значењето на теоремата се менува. А дури и ако важат, гаранцијата не е целосен класичен zero-knowledge; тоа е релаксираната, proof-theoretic верзија од трудот.

Затоа разумната проценка е: висока доверба дека трудот воспоставува кохерентен условен резултат за можност; умерена доверба дека претпоставките го опишуваат криптографскиот свет во кој навистина живееме; и ниска доверба за каква било непосредна практична последица.

# Зошто е важно

Трудот отвора пат што се сметаше за затворен.

Класичната теорија вели: целосен zero-knowledge не може да биде една порака без setup и не може да има perfect soundness. Трудот на Ilango вели: ако ги бараме последиците од zero-knowledge што може да се тестираат во безбедносни игри, и ако дозволиме безбедносната дефиниција да зависи од тоа што еден правилник може или не може ефикасно да побие, тогаш голем дел од корисното однесување може повторно да се добие — со една порака, без setup и со perfect soundness.

Тоа не е мала промена во дефиницијата. Тоа е поинаков начин на размислување за криптографските гаранции. Наместо да прашуваме само што постои, прашуваме што нашиот правилник може да исклучи. Наместо недокажливоста да се третира како филозофска непријатност, се користи како структура.

Практичниот свет можеби нема да се смени утре. Но концептуалната карта се менува. Сега постои формална смисла во која „никој не може ефикасно да докаже дека тајната протекла“ може да биде доволно силно за да врати многу од game-based заштитите што ги сакавме од „тајната не протекла“.

Токму затоа Gödel е во насловот.

# Кратко резиме

Zero-knowledge доказите му овозможуваат на prover да го убеди verifier дека едно тврдење е точно без да го открие сведокот. Класичните резултати за невозможност велат дека zero-knowledge не може да се собере во една порака без setup и не може да има perfect soundness. Трудот на Rahul Ilango не ги побива тие невозможности. Тој дефинира послаб поим, **effectively zero-knowledge**: наместо да бара симулатор навистина да постои, бара избраниот proof system — формален правилник како ZFC — да не може ефикасно да докаже дека симулатор не постои. Под големи претпоставки од криптографијата (неинтерактивни witness-indistinguishable докази) и proof complexity (не постои оптимален proof system), трудот конструира еднопораковни prover-и за NP/SAT, без setup и со perfect soundness, што ги постигнуваат falsifiable, game-based последиците од zero-knowledge својство по својство. Еден prover што би ги покривал сите „природни“ такви својства е дополнително, делумно претпоставено проширување — а покривањето буквално на секое falsifiable својство веројатно е невозможно, бидејќи доказите остануваат повторно употребливи. Резултатот е теоретски и условен, не primitive подготвен за примена, но покажува нов начин proof-theoretic недокажливоста да се користи како криптографски ресурс.

# Проверка без претерување

**Што покажува трудот:** Под наведените претпоставки може да се изградат еднопораковни, без-setup, perfectly sound prover-и за NP/SAT што се effectively zero-knowledge релативно на кој било избран proof system и што ја постигнуваат секоја falsifiable game-based последица од класичниот zero-knowledge.

**Што е веројатно, но не е безусловно докажано:** Дека потребните претпоставки од proof complexity и криптографијата навистина важат. Тие се сериозни и добро проучени претпоставки — а трудот покажува дека се суштински неопходни, не само доволни — но сепак се претпоставки.

**Што не покажува:** Класичен zero-knowledge без интеракција, без setup и со perfect soundness; практичен систем подготвен за примена; deniability или невозможност за повторна употреба на доказите; ниту дека Gödel-овата теорема за нецелосност сама по себе ја обезбедува криптографијата.

**Главни ограничувања:** Гаранцијата е релаксација на zero-knowledge; најширокатa верзија зависи од повеќе претпоставки; тврдењата за еден универзален prover остануваат делумно претпоставени; а резултатот е првенствено фундаментален.

**Колкава доверба треба да има општ читател?** Висока доверба дека ова е важен условен теоретски резултат ако се прифатат дефинициите. Умерена доверба дека претпоставките ја опишуваат реалноста. Ниска доверба за непосредна практична примена. Безбедната поента е: трудот не ги руши невозможностите на zero-knowledge; наоѓа нов proof-theoretic начин да ги заобиколи оние делови од нив што се важни за многу безбедносни игри.

## Извори

Ilango, IACR ePrint 2025/1296
https://eprint.iacr.org/2025/1296

FOCS 2025, DOI 10.1109/FOCS63196.2025.00058
https://doi.org/10.1109/FOCS63196.2025.00058